9/30/99 RADAC meeting 10:00-12:00

I. Donna Smith gave a presentation which is attached to this document

II. Jim Lanzi spoke about
- Filter test bed
- The filter class
- This works well, and would fit in the CORBA standard

III. Tom Taylor
- What type of equipment should be ordered for the filter test bed
- Everyone should look at this system, free thinking is encouraged and ideas are needed
- Talked of suggestions made during audit by S. Godfrey and G. Meyers

III. Comments
- Next meeting - 10/07/99 @ 0900 hours for Tim's CORBA demonstration

---

RADAC Redesign Meeting September 30, 1999 (by Donna Smith)

The old RADAC contained approximately 400,000 lines of code. Using a Software Engineering Lab 70% reuse estimate, the RADAC redesign equates to 240 person-months of effort. We have completed preliminary parts of the re-design. A group will meet Monday October 4th to discuss requirements.

We have made some design decisions and Tim Turner has begun a trade study.

Naval Undersea Warfare Center, Test & Evaluation Department prepared a presentation entitled "Towards a Common Architecture: Real/Synthetic Training T&E Ranges" as part of an R Consortium. The on-line presentation discussed architecture layers for an Open System. Their layering approach included the following:

1. Applications
2. Architecture & Framework
3. Object Libraries
4. Interface
5. Implementation (Industry Standard: GUI FRESCO, Language ANSI C/C++, OS POSIX)
6. Distribution Middleware (Industry Standard: CORBA)
7. Connectivity (Industry Standard: TCP/IP or other standard transport layer)

Starting from the bottom layer, we have identified our connectivity in the requirement document. I believe that Drew Green added the requirement that we use the existing ethernet 10 base T capabilities in the control center.

Tim Turner has been exploring the distribution layer. His CORBA demonstration should take place next week.

As for implementation, the general consensus appears to be C++ for a language, because of its object oriented nature, reasonable reliability (when used with care), and popularity. Ada was another option, but our expertise does not cover Ada.

Also under implementation is the operating system choice. Jim Lanzi, representing the Safety Office, is primarily concerned about reliability. Our top contenders for operating system were Windows NT and UNIX. Safety's choice is UNIX. Susan Semancik advised that we should consider a processor type in connection with the operating system. Low cost microprocessors (PCs) can sometimes match or exceed the cost of larger systems, by requiring more units and more maintenance. A popular PC based UNIX OS is Red Hat's Linux. The ULDB development team has selected Red Hat Linux for their application. So far, Pam Pittman seems pleased with the selection. PC based is a plus, because we are looking for relatively inexpensive systems that we can copy and send multiple places.

The Front-end Processor is a logical first step in the new RADAC. We are running out of NASA built DRT boards, which are used to collect Program Time and radar LTAS and MDDF data. We have discussed a Front-end Processor that can be sent to remote sites or used at Wallops to convert data to and from ethernet/IP, LTAS, MDDF, TIME, NASA36, and Serial formats. The need for a new Front-end is growing. If Wallops built DRT boards are replaced with a commercial board, several real-time programs will need to be modified as the commercial replacements are phased in. RADAC, TELPro, PCGDS, Wind Weighting, Countdown, and other 584 supported programs will need to be re-compiled as the DRT cards are replaced. If we add a Front-end processor, we define the format (probably ethernet) that our software expects. Future changes to data collection cards change only the Front-end processors, not all the other systems. Rodney Davis and Susannah Warner already prepared some plans for a Front-end processor. Use of a Front-end processor is included in the already in progress Wind Weighting update to NT. Plus, Drew Green has mentioned a possible need for converting ethernet to LTAS for a February BQM launch to enable transfer of GPS data from a Navy BQM controller to the RADAC. Jim Lanzi and I would also like to see a Front-end processor started using ethernet and LTAS, but this time with LTAS transformed into ethernet, to allow data transmission to a Filter Test system.

Revising the PC Interface Library (DRT & NASA36) is approximately 8 months of effort. Using the library in a Front-end Processor is approximately 3 more months of effort. These estimates do not include the addition of telemetry and serial data collection capabilities. We have to test and decide on commercial cards (maybe Apogee, ACB3, or Synchronous Digi). We have to re-write library functions and create a generic Front-end processor design that can be expanded to monitor data quality and include at least telemetry and serial data. Plus data needs to flow in two directions.

Computer:
Hardware to accept information
Program to manipulate information (logic built in or loaded by an administrator or user)
Storage for data including the program

Read, execute, read, execute, read, execute,…

In information technology, especially computers and more recently networks, architecture is a term applied to both the process and the outcome of thinking out and specifying the overall structure, logical components, and the logical interrelationships of a computer, its operating system, a network, or other conception. An architecture can be a reference model, such as the Open Systems Interconnection (OSI) reference model, intended as a model for specific product architectures or it can be a specific product architecture, such as that for an Intel Pentium microprocessor or for IBM's OS/390 operating system.

Computer architecture can be divided into five fundamental components: input/output, storage, communication, control, and processing. In practice, each of these components (sometimes called subsystems) is sometimes said to have an architecture, so, as usual, context contributes to usage and meaning.

By comparison, the term design connotes thinking that has less scope than architecture. An architecture is a design, but most designs are not architectures. A single component or a new function has a design that has to fit within the overall architecture.

A similar term, framework, can be thought of as the structural part of an architecture.

Naval Undersea Warfare Center
Test & Evaluation Department
Presentation: Towards a Common Architecture: Real/Synthetic Training T&E Ranges
R Consortium

Applications
Architecture & Framework
Object Libraries
Interface
Implementation (Industry Standard: GUI FRESCO, Language ANSI C/C++, OS POSIX)
Distribution Middleware (Industry Standard: CORBA)
Connectivity (Industry Standard: TCP/IP or other standard transport layer)

Our connectivity has been defined as the 10baseT ethernet
Our middleware appears to be leaning towared CORBA with Tim Turner's research
Languages suggested included Ada & C/C++. We seem to have settled on C++ based on our knowledge. The top Operating System considerations seem to be UNIX and NT. Operating System is related to the choice of processor. We seem to be leaning towards PC (microprocessors) – cheap & scalable. Versions of UNIX run on PCs. Safety is against the use of NT because of reliability issues. If we choose a version of UNIX on a PC, Linux is popular & the the ULDB development group is already using the Red Hat Linux Operating System. Red Hat Linux is described on its packaging as POSIX based.
The Graphical User Interface…

POSIX (Portable Operating System Interface) is a set of standard operating system interfaces based on the UNIX operating system. The need for standardization arose because enterprises using computers wanted to be able to develop programs which could be moved among different manufacturer's computer systems without having to be recoded. UNIX was selected as the basis for a standard system interface partly because it was "manufacturer-neutral." However, several major versions of UNIX existed so there was a need to develop a common denominator system.

Informally, each standard in the POSIX set is defined by a decimal following the POSIX. Thus, POSIX.1 is the standard for an application program interface in the C language. POSIX.2 is the standard shell and utilities interface (that is to say, the user's command interface with the operating system). These are the main two interfaces, but additional interfaces, such as POSIX.4 for thread management, have been developed or are being developed. The POSIX interfaces were developed under the auspices of the IEEE.

Recently, POSIX.1 and POSIX.2 interfaces were included into a somewhat larger interface known as the X/Open Programming Guide 4.2 (also known as the "Single UNIX Specification" and "UNIX 95"). The Open Group, an industry standards group, owns the UNIX trademark and can thus "brand" operating systems that conform to the interface as "UNIX" systems. IBM's OS/390 is an example of an operating system that includes a branded UNIX interface.

From www.whatis.com

Approximately 2% of the RADAC redesign is complete.